Zoea

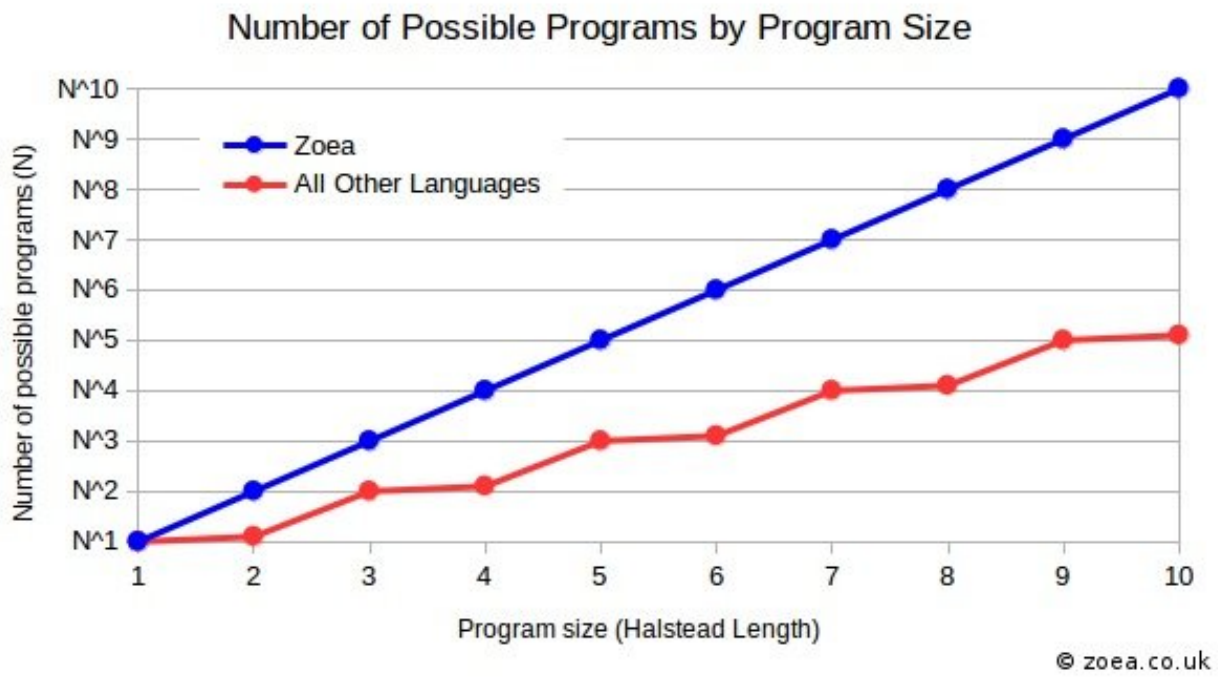
Research Note 9

The Expressive Power of Zoea

Edward McDaid & Sarah McDaid

05 Dec 2020

Zoea is an AI that lets anyone who can describe a problem as a set of input and output examples produce real code. Some might ask can Zoea really produce all the programs that a conventional language can just by writing test cases? The short answer is yes.



One way we can compare Zoea with other programming languages is to consider programs of different sizes and for each size calculate how many possible different programs we could theoretically produce. If conventional programming languages could specify more programs of a given length than Zoea then we would have to write ever

larger numbers of test cases for each incremental increase in conventional program size. This is not the case.

We can demonstrate this by breaking programs down to their fundamental elements. Programs in any programming language are composed of operators and operands. Operators are the syntactic and semantic elements of the language such as '+' and 'print'. Operands represent the data and include the variables and constant values that occur within a program. If we count all of the operators and operands in a given program then we get a figure called the Halstead length. This is simply a measure of how many individual language elements there are and this is the metric that we will use.

To understand how many possible programs of a given size we could produce we need to consider how many different operators, variables and constants we can have as well as the range of values that each of these categories can assume. Different programming languages have different numbers of operators. Most programming languages provide a few hundred operators including keywords and built in functions. The number of operators could be up to a few thousand if we include the standard libraries in larger programming languages.

Similarly different programs have different numbers of variables. Larger programs tend to have more variables and the choice of programming language can also impact their frequency. It is always the case that the number of variables in a program is less than the program length.

The range of values that constants can assume is many orders of magnitude larger than the number of operators or variables. This is basically all the values that can be represented by a computer including numbers and strings as well as composite data structures like arrays and objects. In theory this number is infinite but in reality it is effectively limited by storage. Nevertheless it is an extremely large number. For convenience we will refer to it as N. This allows us to make relative comparisons between numbers of programs expressed as multiples of N.

Understanding the relative impact of operator, variable and constant count on the number of possible programs allows us to produce relative estimates for different program lengths. (See chart.) These figures are all expressed in terms of N. It is clear that the number of

constants dominates the results. The step function shape comes from the fact that valid programs in conventional languages of length > 1 cannot be composed entirely from constants. The flat part of each step is in fact a relatively small increase that corresponds to the contribution from variables and operators in addition to constants. It is worth noting that the figures for conventional languages represent an upper bound. It is very unusual to have a program in a conventional language that consists mainly of constants.

Zoea programs on the other hand are composed almost entirely of constant values. These include inputs and outputs for each test case as well as any number of intermediate values. For example a Zoea program with 12 values might correspond to six test cases with a single input and output value, or it could be 4 test cases with two inputs and an output. In any event the number of possible Zoea programs of a given length is N to the power of that length.

The graph can also be used to estimate an upper bound in terms of how many Zoea values we might need to represent a program of a given size. For example a program size of 9 gives N^5 possible conventional programs. This corresponds to a Zoea program containing 5 values.

It is clear that for any given program size > 1 Zoea can produce more possible programs than a conventional language. Also as the length increases this difference gets ever larger. Remember the data shows the upper bound for conventional programs so in the real world the red line would be much closer to the X axis. Also the chart is logarithmic so the gap between the two series is really increasing exponentially.

This shows that Zoea can in fact produce as many programs as conventional languages and that the number of test cases required does not grow exponentially with increasing program size.

Learn more at **zoea.co.uk**